Applications Frameworks

Theory and Practice

Ryan Somma

CIS 518

Evolution of Programming

Binary=>Assembly=>Procedural

Instruction Inflation

Object-Oriented Encapsulation

- Trend toward reducing complexity
- Trend toward improving productivity

Frameworks Continue These Trends

Framework Scopes

- Entire application (Application Framework)
- Or Sub-Component (ie. Template Engines, Sematic Web, JavaScript)

Framework Advantages

- Reusable solutions (DRY Principle)
- Base Code set provides a jump start to applications development (RAD SDLC)

Framework Design Patterns

- Are frameworks just one big design pattern?
- Frameworks and design patterns exist at different levels of abstraction.
- Frameworks are collections of interacting patterns.

Framework Design Patterns

- **Inversion of Control (IoC):** the framework references the programmer's code, not the programmer the framework.
- **Front Controller:** there is a single point in the application for handling application requests.

Framework Design Patterns

- **Model View Controller (MVC):** separate design, request processing, and domain functionality.
- **Object-Relational Mapper (ORM):** transforms data stored in relational tables into domain objects.

Strengths and Weakness

- **Extensibility:** does the framework use blackbox or whitebox reuse to expand its functionality?
- **Integrateability:** can the framework interface with a wide variety of other applications and services?
- **Metrics:** how will the framework's architecture affect design measurements and efficiency of the application?
- **Maturity:** how long has the framework been around, and how stable is it as a result?

Strengths and Weaknesses

- **Learning Curve:** how much time must be invested to understand and begin producing product with the framework in relation to how many features the framework provides?
- **Documentation and Support:** is there thorough documentation and an active user community to draw on as resources?

“Make or Buy?” Analysis

- Adopting an Existing Framework
  - Established, peer-reviewed code
  - No “reinventing the wheel.”
  - Community resources.
  - Cost effective

“Make or Buy?” Analysis

- Developing a Framework in House
  - Total customization
  - Streamlined functionality, improved performance
  - No need for backward compatibility, adopt the latest innovations

“Make or Buy?” Analysis

- Organization-Dependent Considerations
  - Is the Organization capable of producing a high-quality framework (resources, institutional knowledge)?
  - Avoid “Not invented here” syndrome
  - Does the existing solution meet enough criteria to make an in-house solution illogical?

Conclusions

- Frameworks are crucial to organization information systems development
- Consideration to maturity, extensibility, and integrateability should be taken when choosing a framework
- Developing a framework should include adherence to quality and established framework design patterns.